

A Deterministic Quantum Hash Framework for Multi-Scale Physical and Informational Systems

Abstract

A deterministic framework is defined for constructing a quantum hash as a hybrid physical-informational descriptor applicable across multiple scales, from macroscopic physical systems to biological structures.[file:202] To make the framework reproducible in computational, physical, and biological settings, each theoretical component is translated into observable dimensions, candidate indicators, and explicit normalization rules that support deterministic implementation.[file:202] The resulting operational specification preserves the original structure of the model while clarifying how state observables, entropy, energy scale, temporal indexing, causal consistency, and metadata are measured, standardized, and encoded before cryptographic hashing.[file:202]

1. Introduction

The increasing integration of physical modeling, simulation, and data-driven workflows requires reproducible methods for state identification and traceability across heterogeneous systems.[file:202] A deterministic hash framework is useful only if its theoretical terms can be translated into operational variables that can be measured, normalized, and serialized in a stable way across implementations.[file:202]

The present formulation therefore defines the quantum hash not only as a conceptual mapping, but also as an operational pipeline from observables to indicators, from indicators to normalized components, and from normalized components to a deterministic cryptographic identifier.[file:202]

2. Operational Form of the Framework

2.1 Core Definition

The operational quantum hash is defined by

$$H_q(O; t) := \mathcal{H}(X_o(t), S(\rho_o(t)), E_o, T(t), \mathcal{C}(t), \mathcal{M}_o),$$

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.
 No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.
 See full “License and Conditions of Use” notice for details.
 In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.
 Author: Carlos Daniel Borrego Romero

where

$$X_o(t) := \text{Tr}(\rho_o(t)\hat{H}_{\text{eff}}(t))$$

is the scalar observable summary derived from the effective operator.[file:202] The operational challenge is to define how each quantity is observed or constructed in practice.[file:202]

2.2 Operational Components

The framework is implemented through six operational components:[file:202]

- State representation $\rho_o(t)$.
- Effective observable summary $X_o(t)$.
- Entropy descriptor $S(\rho_o(t))$ and normalized entropy S_{norm} .
- Energy-scale descriptor E_o .
- Temporal index $T(t)$.
- Causal and metadata descriptors $\mathcal{C}(t)$ and $\mathcal{M}_o$. [file:202]

Each of these components is translated below into observable dimensions, candidate indicators, and normalization rules.[file:202]

3. Translation to Observable Dimensions

3.1 State Representation $\rho_o(t)$

The theoretical density operator is operationally interpreted as a normalized state representation of the system under study.[file:202] In fully quantum systems, candidate indicators include experimentally reconstructed density matrices, simulated density matrices, reduced density operators, or probability amplitudes transformed into a density representation.[file:202] In classical or hybrid systems, the corresponding operational substitute is a normalized state distribution or structured probability vector over the relevant state space, provided it satisfies positivity and normalization constraints.[file:202]

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.
 No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.
 See full “License and Conditions of Use” notice for details.
 In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.
 Author: Carlos Daniel Borrego Romero

The observable dimension for this component is therefore **state configuration and probabilistic structure**.^[file:202] Candidate indicators include:

- Quantum tomography outputs.
- Numerical simulation state vectors transformed into ρ .
- Reduced density matrices for subsystems.
- Classical probability distributions over discrete states.
- Normalized feature-state vectors in informational systems.^[file:202]

3.2 Effective Operator Summary $X_o(t) = \text{Tr}(\rho_o \hat{H}_{\text{eff}})$

This component is operationally interpreted as the principal scalar summary of system structure and dynamics extracted through the effective operator.^[file:202] The observable dimension is **expected system response under an effective structural-dynamical operator**.^[file:202]

Candidate indicators depend on system type:^[file:202]

- Hamiltonian expectation values in simulated or measured quantum systems.
- Effective field-energy summaries in macroscopic physical models.
- Structured operator scores derived from interaction matrices, correlation tensors, or graph operators.
- Weighted observable composites for complex informational systems.
- Domain-specific surrogate expectation scores for systems where an explicit operator is approximated computationally.^[file:202]

3.3 Entropy Descriptor $S(\rho_o)$

The entropy term is operationally interpreted as the informational disorder or configurational uncertainty of the system.^[file:202] The observable dimension is **informational complexity of the state representation**.^[file:202]

Candidate indicators include:^[file:202]

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.

No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.

See full “License and Conditions of Use” notice for details.

In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.

Author: Carlos Daniel Borrego Romero

- Von Neumann entropy for density operators.
- Shannon entropy for classical probability distributions.
- Spectral entropy derived from eigenvalue distributions.
- Reduced subsystem entropy for composite systems.
- Entropy surrogates based on empirical occupancy distributions or state-frequency profiles.[file:202]

3.4 Energy Scale E_o

The energy-scale term is operationally interpreted as the macroscopic or effective scale marker that anchors the descriptor to the physical magnitude of the system.[file:202] The observable dimension is **system scale through energy or mass-energy equivalence**. [file:202]

Candidate indicators include:[file:202]

- Direct mass measurement with $E = Mc^2$.
- Effective inertial mass for modeled systems.
- Equivalent rest-energy scale from astrophysical or physical catalogs.
- Effective energy proxies for nonliteral physical systems, provided the mapping is formally declared and consistently applied.[file:202]

3.5 Temporal Index $T(t)$

The temporal component is operationally interpreted as a deterministic discrete label for state occurrence or evaluation.[file:202] The observable dimension is **time-position within a reproducible sequence**. [file:202]

Candidate indicators include:[file:202]

- Timestamp rounded to a fixed evaluation interval.
- Simulation step number.
- Acquisition cycle identifier.

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.

No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.

See full “License and Conditions of Use” notice for details.

In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.

Author: Carlos Daniel Borrego Romero

- Visit number or batch index.
- Event sequence counter.[file:202]

3.6 Causal Descriptor $\mathcal{C}(t)$

The causal component is operationally interpreted as a deterministic descriptor of admissible ordering, separation, or lineage consistency.[file:202] The observable dimension is **causal or process-validity relation between states, events, or entities**. [file:202]

Candidate indicators include:[file:202]

- Binary causal admissibility flags.
- Temporal separation thresholds.
- Lineage consistency codes.
- Dependency-validity indicators in simulation pipelines.
- Directed acyclic graph edge constraints in workflow systems.[file:202]

3.7 Structured Metadata $\mathcal{M}_o$

The metadata layer is operationally interpreted as the structured contextual information required to make the hash reproducible and uniquely attributable.[file:202] The observable dimension is **identity and provenance context of the evaluated system state**. [file:202]

Candidate indicators include:[file:202]

- System class.
- Scale class.
- Protocol identifier.
- Model version.
- Device or software version.
- Experiment identifier.
- Biological tissue or sample type.[file:202]

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.
 No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.
 See full “License and Conditions of Use” notice for details.
 In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.
 Author: Carlos Daniel Borrego Romero

4. Candidate Indicator Architecture

4.1 Component-Level Mapping Table

Component	Observable dimension	Candidate indicators	Default orientation
$\rho_o(t)$	State configuration and probabilistic structure	Density matrix, reduced density matrix, normalized probability vector, simulation state profile	Higher values are structural entries, not risk-oriented
$X_o(t)$	Effective observable summary	Hamiltonian expectation, operator score, field-energy summary, graph-operator expectation	Signed or nonnegative according to operator definition
$S(\rho_o)$	Informational complexity	Von Neumann entropy, Shannon entropy, spectral entropy, occupancy entropy	Higher = more disorder
S_{norm}	Bounded entropy position	Entropy divided by class-specific S_{max}	Higher = more normalized disorder
E_o	Physical or effective scale	Mass-energy, effective energy proxy, catalog energy scale	Higher = larger macroscopic scale
$T(t)$	Time position	Step index, rounded timestamp, acquisition cycle, event counter	Ordered increasing index
$\mathcal{C}(t)$	Causal consistency	Binary flag, bounded penalty, lineage code, dependency-validity code	Higher may mean stronger admissibility if coded on [0,1]
$\mathcal{M}_o$	Provenance context	System ID, version, protocol, device, sample class, pipeline ID	Categorical or coded fields

This mapping makes each theoretical component operationally measurable while preserving the model structure.[file:202]

4.2 Derived Continuous Components

The model defines the following derived quantities:[file:202]

$$S_{\text{norm}}(\rho_o) := \frac{S(\rho_o)}{S_{\text{max}}},$$

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.
 No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.
 See full “License and Conditions of Use” notice for details.
 In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.
 Author: Carlos Daniel Borrego Romero

$$\text{Correction}(O; t) := \kappa(S_{\text{norm}}(\rho_O(t)) - S_{\text{ref}}),$$

$$\text{Re}(H_q^{\text{cont}}) = \log_{10}(E_O) + \frac{\text{Correction}(O; t)}{100},$$

$$\text{Im}(H_q^{\text{cont}}) = \gamma S_{\text{norm}}(\rho_O(t)).$$

These quantities are not directly measured; they are computed from the operationalized base components after normalization and calibration.[file:202]

5. Scales and Normalization Rules

5.1 General Normalization Principle

Every indicator entering the framework must either already live on a mathematically valid domain or be transformed into one before computation.[file:202] The operational objective is not to force all variables into the same scale, but to ensure that each variable enters its corresponding formula on a valid and reproducible domain.[file:202]

5.2 State Representation Rules

The operational state representation must satisfy:

$$\rho_O \geq 0, \rho_O = \rho_O^\dagger, \text{Tr}(\rho_O) = 1.$$

For empirical matrices estimated from noisy data, the normalization procedure should include:[file:202]

1. Symmetrization, $\rho \leftarrow (\rho + \rho^\dagger)/2$.
2. Projection to the positive semidefinite cone if small negative eigenvalues appear from numerical noise.
3. Trace normalization, $\rho \leftarrow \rho/\text{Tr}(\rho)$. [file:202]

For classical state vectors, probabilities must be clipped at zero and renormalized to sum to one.[file:202]

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.
 No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.
 See full “License and Conditions of Use” notice for details.
 In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.
 Author: Carlos Daniel Borrego Romero

5.3 Effective Observable Scaling Rules

The expectation value $X_O(t) = \text{Tr}(\rho_O \hat{H}_{\text{eff}})$ does not need to be constrained to $[0,1]$ unless a specific downstream implementation requires bounded internal scaling.[file:202] However, comparability across systems improves if the effective operator is calibrated within each system class.[file:202]

Recommended options are:[file:202]

- **Native physical scale** when operator units are intrinsically meaningful.
- **Reference-centered standardization** within a system class:

$$X_O^* = \frac{X_O - \mu_X}{\sigma_X}$$

- **Bounded logistic transform** when a class-specific bounded representation is needed:

$$\tilde{X}_O = \frac{1}{1 + e^{-X_O^*}}.$$

5.4 Entropy Normalization Rules

The entropy term must be normalized using a class-specific $S_{\text{max}} > 0$, so that

$$S_{\text{norm}} = \frac{S}{S_{\text{max}}} \in [0,1].$$

Recommended operational choices include:[file:202]

- $S_{\text{max}} = \log d$ for finite-dimensional quantum systems.
- The empirical maximum entropy over a predefined reference class.
- A theoretical class-specific maximum derived from constrained state-space assumptions.[file:202]

Values outside $[0,1]$ after normalization indicate inconsistency between the entropy estimator and the chosen reference scale and should trigger recalibration.[file:202]

5.5 Energy Scaling Rules

Energy enters the real part through $\log_{10}(E_O)$, so the only required operational restriction is

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.
No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.
See full “License and Conditions of Use” notice for details.
In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.
Author: Carlos Daniel Borrego Romero

$$E_o > 0.$$

For direct mass-based systems, energy is computed via $E_o = M_o c^2$. [file:202] For systems using an effective scale surrogate, the mapping to a positive effective energy variable must be explicitly declared and fixed across all evaluations. [file:202] If inter-class comparison is needed beyond raw $\log_{10}(E_o)$, a secondary class-normalized quantity may be stored in parallel, but the primary model should preserve the native macroscopic scaling. [file:202]

5.6 Temporal Index Rules

The time index is discrete and should be normalized by deterministic rounding rather than by statistical scaling. [file:202] The standard rule is

$$T(t) = T_0 + n\Delta t, n \in \mathbb{Z}, \Delta t > 0,$$

with a fixed choice of time unit, origin, and rounding rule for all evaluations in the same system class. [file:202]

5.7 Causal Descriptor Rules

The causal component should be normalized to a finite deterministic code space. [file:202] Suitable rules include: [file:202]

- **Binary rule:** $\mathcal{C} \in \{0,1\}$, where 1 means causally admissible.
- **Bounded penalty rule:** $\mathcal{C} \in [0,1]$, where 1 means perfect consistency and smaller values indicate increasing violation.
- **Ordinal code rule:** a finite set such as $\{0,1,2\}$ representing invalid, uncertain, and valid causal relation. [file:202]

The selected rule must remain fixed within the implementation because the hash is sensitive to representation changes. [file:202]

5.8 Metadata Encoding Rules

Metadata are not normalized statistically but standardized structurally. [file:202] The required rules are: [file:202]

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.

No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.

See full “License and Conditions of Use” notice for details.

In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.

Author: Carlos Daniel Borrego Romero

1. Fixed field names.
2. Fixed ordering.
3. Controlled vocabularies for categorical fields.
4. Fixed numeric precision for continuous metadata.
5. Stable serialization format such as canonical JSON or deterministic delimited strings.[file:202]

6. Operational Workflow

6.1 Measurement and Construction Pipeline

A practical implementation follows this sequence:[file:202]

1. Acquire or construct the system state representation.
2. Validate and normalize the state object.
3. Define or estimate the effective operator for the relevant system class.
4. Compute the scalar expectation $X_O(t)$.
5. Compute entropy and normalized entropy.
6. Compute the energy-scale descriptor.
7. Assign the discrete temporal index.
8. Encode the causal descriptor.
9. Assemble structured metadata.
10. Compute the continuous descriptor.
11. Serialize the full canonical input.
12. Generate the final cryptographic hash.[file:202]

6.2 Minimal, Intermediate, and Expanded Operational Modes

A **minimal mode** can be implemented with a normalized state representation, a class-specific effective operator, entropy, energy scale, time index, and fixed metadata.[file:202] An **intermediate mode** adds richer causal coding, calibrated operator decomposition, and class-specific normalization

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.
 No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.
 See full “License and Conditions of Use” notice for details.
 In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.
 Author: Carlos Daniel Borrego Romero

tables.[file:202] An **expanded mode** includes domain-specific suboperators, biological lineage constraints, multi-source provenance metadata, and advanced class-specific entropy references.[file:202]

7. Biological Operational Extension

For biological systems, the operationalized specialized descriptor is

$$H_q^{\text{bio}} = \mathcal{H}(E_{\text{bio}}, S_{\text{bio}}, T, \text{bio-parameters}, \mathcal{C}_{\text{bio}}).$$

The corresponding observable dimensions are:[file:202]

- **Biological scale descriptor:** effective tissue or sample scale.
- **Biological entropy descriptor:** heterogeneity or informational disorder of the biological state.
- **Bio-parameters:** tissue type, protocol, preparation condition, batch, assay type.
- **Biological causal descriptor:** lineage, sampling sequence, or intervention relation.[file:202]

Candidate indicators include tissue mass proxies, omics-derived entropy surrogates, aggregated morphology states, protocol identifiers, and lineage-validity flags, provided the framework is restricted to aggregated, non-sensitive descriptors rather than raw patient-level or genomic content.[file:202]

8. Reporting Standard

Each operational implementation should report at least the following:[file:202]

- State representation type.
- Effective operator family.
- Entropy estimator used.
- Choice of S_{max} .
- Energy definition or surrogate.
- Time-index rule.
- Causal-coding rule.

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.

No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.

See full “License and Conditions of Use” notice for details.

In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.

Author: Carlos Daniel Borrego Romero

- Metadata schema.
- Serialization format.
- Cryptographic hash function used.[file:202]

This reporting standard is essential for reproducibility because different operational choices can produce different deterministic hashes even when the conceptual framework remains unchanged.[file:202]

9. Discussion

The operational formulation converts the quantum hash from a conceptual descriptor into a reproducible computational pipeline.[file:202] Its main strength is the explicit separation between measured state structure, derived informational quantities, scaling terms, deterministic indexing, and cryptographic serialization.[file:202]

The model remains flexible because the precise indicators can vary by system class, but that flexibility is controlled by fixed normalization and encoding rules within each class.[file:202] This preserves interpretability while allowing adaptation across physical, informational, and biological domains.[file:202]

10. Conclusion

The deterministic quantum hash framework can be implemented operationally by translating each theoretical component into observable dimensions, candidate indicators, and explicit normalization rules.[file:202] The resulting formulation preserves the physical-informational logic of the model while making it suitable for reproducible deployment in simulations, data systems, multi-scale physical analysis, and biological provenance workflows.[file:202]

LICENSE AND CONDITIONS OF USE.

The content is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license:

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

By using this content and model, you agree to:

Provide appropriate credit, include a link to the above license, and indicate if changes were made.

Not use the material or any derivative works for commercial purposes.

Distribute any adaptations or derivative works only under the same CC BY-NC-SA 4.0 license.

Not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.

No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.

See full “License and Conditions of Use” notice for details.

In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.

Author: Carlos Daniel Borrego Romero

Data logging and trade secrets policy.

The design, implementation, and deployment of this model, as well as any derivatives, do not authorize the collection, storage, or exploitation of IP addresses, unique device identifiers, or other technical traceability data for commercial profiling, targeted advertising, or any other economic exploitation.

Any system implementing this model must limit the logging and retention of IP addresses and other identifiable data strictly to what is necessary to:

(a) maintain technical security and service integrity; and

(b) comply with applicable legal obligations (for example, any minimum log retention required by law).

It is prohibited to use this model or its derivatives to:

capture, store, or exploit trade secrets, proprietary know-how, or confidential information of users or third parties for any purpose other than providing the technical service described in this documentation; or train or improve closed, commercial systems using data that contains confidential information of third parties without their prior, explicit, written consent.

---+EOD+---